BIGDATA Processing with Hadoop
(MR Programming)

Problem 1: Sort the file in descending order

Write a MR program that sorts the numbers of given input file in descending order. Assume that each line of file consists of single number only.

Problem 2: Finding Distinct Lines

Given the input file, write an MR program that removes duplicate lines after converting to lower case.

Problem 3: Computing Matrix Transpose

Write an MR program that takes as an input a large file containing a matrix and returns an output
file that has it transposed. The final result can be either a single file or multiple files (in which case, when concatenated, they should give the same result as a single file).

For example, the input matrix:

1 2 3 4
5 6 7 8
9 10 11 12

should be transposed to:

1 5 9
2 6 10
3 7 11
4 8 12

Problem 4: Inverted Index Construction

Given a set of documents, write an MR program to build an inverted index. An inverted index is a dictionary where each word is associated with a list of the document identifiers in which that word appears. Use the files of given input folder as input and produce an inverted index.
For instance, given the following documents:
d1.txt: cat dog cat fox
d2.txt: cat bear cat cat fox
d3.txt: fox wolf dog

You should build the following full inverted index:
bear:1:(d2.txt,1)
cat:2:(d1.txt, 2), (d2.txt, 3)

dog:2:(d1.txt, 1), (d3.txt, 1)
fox:3:(d1.txt, 1), (d2.txt, 1), (d3.txt, 1)
wolf:1:(d2.txt,1)

For each term (word), there is a single record consisting of a number and a list of what are termed postings; the semicolon character (':') is used to delimit the fields of each record. The first field is a number that represents the number of documents that contain the term. Then a list of postings follows where each posting is a pair consisting of the document name and the frequency of the word in that specific document. Note that terms are sorted alphabetically and also that the items inside lists are also sorted alphabetically by document identifier. For example, the following line:

cat:2:(d1.txt, 2), (d2.txt, 3)

indicates that the word cat appears in two documents, two times in document d1.txt and three times in document d2.txt. Use the document name for identifying a document.

Problem 5: Basic Last.fm Statistics

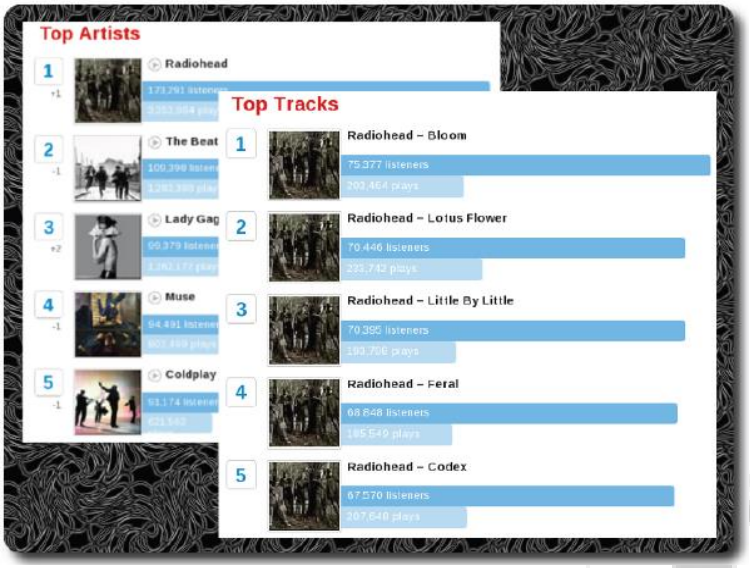
Last.fm is an Internet radio and community-driven music discovery service founded in 2002. Users transmit information to Last.fm servers indicating which songs they are listening to. The received data is processed and stored so the user can access it in the form of charts and so Last.fm can make intelligent taste and compatibility decisions for generating recommendations and radio stations.

The track listening data is obtained from one of two sources:

a) The listen is a scrobble when a user plays a track of his or her own and sends the information to Last.fm through a client application.
b) The listen is a radio listen when the user tunes into a Last.fm radio station and streams a song.
c) Last.fm applications allow users to love, skip or ban each track they listen to. This track listening data is also transmitted to the server.

Write a MR program that takes the incoming listening data and summarize it into a format that can be used to display on the website. Here is a sample screenshot of UI:

BIGDATA Processing with Hadoop
(MR Programming)



Input: (user id, track id, scrobble, radio play, skip), where the last three fields are 0 or 1.

UserId	TrackId	Scrobble	Radio	Skip
111115	222	0	1	0
111113	225	1	0	0
111117	223	0	1	1
111115	225	1	0	0

Output: Unique number of listeners, accumulated listens, scrobbles, radio listens and skips per track

TrackId	numListeners	numPlays	numScrobbles	numRadioPlays	numSkips
222	1	1	0	1	0
223	1	1	0	1	1
225	2	2	2	0	0

You can find sample dataset at this location:
http://www.iro.umontreal.ca/~lisa/datasets/profiledata_06-May-2005.tar.gz